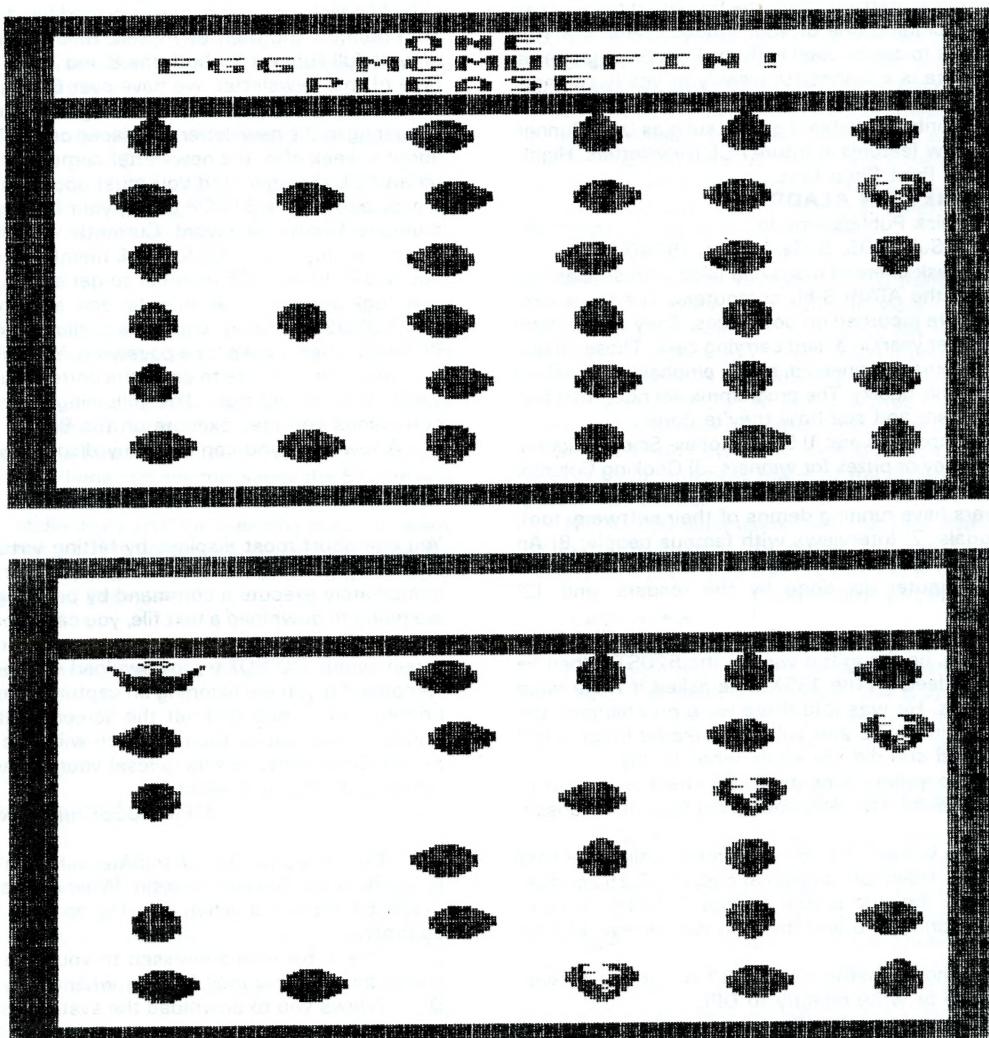


3662 Vine Maple Dr. Eugene OR 97405

JULY - AUGUST 1986

Mike Dunn, Jim Bumpas, Larry Gold, co-editors



CHARLIE CHICKEN

CHARLIE CHICKEN
CHARLIE CHICKEN

News and Reviews

by Mike Dunn, Co-Editor

In this issue, we will feature a new, very original Stan Ockers Game, Charlie Chicken Adventure #1. This is a wonderful game for children, written in Action. It will be on our next "Best of ACE" disk in runtime form for those of you who do not have an Action! cartridge.

Next issue will be very different. We are making many new changes in the Newsletter, and will be doing one or more of the following: Bulk Mail, larger paper size, different size print, etc. These changes are to save money for now and allow for future expansion if needed.

For the month of August, my family and I will be visiting England and Scotland, so write Jim Bumpas if you need anything. He will also pick up my ACE mail.

We have received several new programs for review this month. For the ST, **Typesetter ST** and **Rubber Stamp ST** from Xlent Software, POB 5228, Springfield, VA 22150. These are currently under review and will be extensively reported on in the next issue. **The Print Shop Companion** (Broderbund, 17 Paul Dr., San Rafael, CA 94903-2101, \$35) is a new utility disk in the Print Shop series for the 8-bit computers. It only works on XL or XE machines or on machines with 64K. It does not work on regular 400 or 800's. It is a very easy to use and wonderful program allowing you to greatly increase the usefulness of Print Shop. There are three Editors: a Graphic +, a Border Editor and a Font Editor allowing you to change, draw and superimpose graphics. There is a new, powerful drawing program to make new graphics, and use numerical graphics. With the Border Editor, you can make various borders including one of your initials. There is a Title Maker and Creature Maker to be used with the Print Shop graphics. If that's not enough, there is a wonderful weekly or yearly calendar program as well as 12 new fonts, 50 new borders, 24 numeric graphics, and 20 graphics from favorite Broderbund games such as Lode Runner. We will be using these new features in future ACE Newsletters. Highly recommended to all you Print Shop fans.

THE NEW ALADDIN

Disk Publications Inc

5757 Alpha Road, Suite 105, Dallas, Texas 75240-9990

The New Aladdin is disk-oriented magazine filled with articles and programs about and for the ATARI 8-bit computers. The three jam-packed magazine disks are recorded on both sides. They come every other month (six issues per year) in a hard carrying case. These unique disks contain full color, human engineered, highly emphasized graphics and are close to photographic quality. The programs are not protected. You can break in, list them and see how they're done.

The summary of the contents are: 1) Short stories- Science fiction, etc.; 2) Quizzes with money or prizes for winners; 3) Cooking Column; 4) Movie reviews; 5) Hardware/Software Reviews with running demos of software (advertisers have running demos of their software, too); 5) Cartoons; 6) Editorials; 7) Interviews with famous people; 8) An opinion column 9) Quality game; 10) Question and Answer column; 11) Art gallery for computer art done by the readers; and 12) Educational material.

The documentation says when Jack Tramiel saw the THE NEW ALADDIN demonstration, he thought it was on the 520ST. When he found out that it was indeed on the 130XE, he asked if there were any changes in the chips. He was told there were no changes, the publication was written in BASIC and will be marketed under \$100 per year. He was amazed and did not know what to say.

How do I get a subscription? Send a \$79.95 check made out to Disk Publications, Inc. and send it to: Nora Young, c/o ACE, 105 Hansen, Eugene, OR 97404.

The single issue price will be \$24.95 from a retail outlet. The cost breaks down to 18 double-sided (36 singles) at cost of \$2.25 per disk. Also by subscribing thru the club before August 1, 1986, you can receive 7 issues for the price of 6 and the club will receive \$10 for each subscription.

I am sorry I do not know whether or not DPI will ship overseas. Either drop a note to me or write directly to DPI.

PS. DPI says subscriptions to 8-bit computers will be converted to an ST version as soon as it is introduced for those subscribers who choose to do so at no cost. The ST Aladdin should be out by the end of the year.

MOVIE MAKER

Movie Maker is a program for creating "movies" or cartoons which may be viewed from Movie Maker itself, or sent over a modem. This program was used to create and edit most of the movies on the ACE BBS. When creating or editing a movie, Movie Maker will store every key stroke you make. If you make a mistake, you can use START to backup, or OPTION to go forward over previously entered keys. SELECT will return you to the main menu. You can view your movie at 300 or 1200 baud, or as fast as it can be sent to your screen. To edit an existing movie, load it in, and press

C

for create. Now use the OPTION key to step forward through the movie. If you hold OPTION, the first 10 keys will be displayed at normal speed, then they will be displayed at high speed until you release the key. By using START and OPTION you can change a keystroke entered, but unless it's at the end of the movie, you can only overwrite it. You will not be able to insert or completely remove any keystroke.

Movie Maker takes good advantage of DVC/65's small code size - it gives you over 29K of buffer space. There is one function you may want to use in other programs: opnfile(). This function will normalize the filename you give it with the default drive and the extension you give it, then attempt to open it in the mode you give it. If successful, it returns the iocb number, else it informs the user of a bad open, waits for a keypress, then returns a zero. This is a commonly used function in programs that will be dealing with files.

The ACE BBS uses a program called PD/M. This program used to be Mindlink, but was extensively modified first by the original author, Rich Renner, and then by myself, Ralph Walden. I have set it up to provide full support for both the 8 and 16 bit Atari computers. At the time of this newsletter, we have over 6 megabytes of files available for downloading, and are continuing to expand. All of the programs appearing in the newsletter are placed on the BBS for you to download about a week after the newsletter comes out. To get them, you must be an ACE member, and you must apply for a password by leaving a message for the SYSOP giving your full name, phone number, and a unique 4 letter password. Currently we have about 1200 callers a month, so the best time for ACE members to call is Saturday when you MUST be an ACE member to get online. If you just want to call and look around, feel free to call anytime except Saturday, at 300/1200/2400 baud. Guests are allowed on the board by hitting RETURN when it asks for a password. You will then be asked for your screen width - be sure to enter the correct number as this affects the word wrap on all text. The following is a list of all the possible commands you may execute on the BBS.

A few tips: you can stop any display by hitting control

S

and continue again by hitting control

Q

You can abort most displays by hitting virtually any key. If you are displaying a menu or a message, you can stop the display and immediately execute a command by pressing the desired key. If you are going to download a text file, you can chose not to use X-modem, at which time you may chose to have word wrap used for your current screen width. DO NOT try to download ANY binary/saved files without X-modem! If you are planning on capturing text and sending it to your printer, you should first set the screen width to the width of your printer. Then capture the file which will have word wrap set for your printer. Once done, be sure to reset your screen width as this is always saved with your password.

MENU DESCRIPTION

A — Resets translation (Ascii/Atasci) Atari's normally use Atasci.
B — Read the System bulletin. Whenever a new bulletin is put up, it will be displayed when you log on. You can reread it with this command.

C — Check for mail addressed to you. The BBS will automatically check for any new mail for you whenever you log on.

D — Allows you to download the system help files. There are some additional download files here including the ACE library list. This list contains a description of all the disks available from the ACE library including most of the files found on the BBS.

E — Enter a message into the current message base. You may enter up to 20 lines; each line may contain up to 126 characters. If you press

at the beginning of a line, you will go to the message editor. You can also execute any of the message editor commands by beginning a line with a / . Message commands:

/T Go to top of file and re-enter the message.
/Un Go up n lines from the current line number.
/Dn Go down n lines from the current line number.
/Gn Goto line n and re-enter the line.

/L List the message.
/S Save the message
/A Abort the message

Examples:

/G10 would goto line 10
/U5 would go up 5 lines
/D10 would go down 10 lines

If you make a mistake on a line, you can use /Gn which will list out the line as you entered it. Next retype the line and then on the following line type /G and the number one past the last line you have entered.

Once a message is saved, all RETURNS are removed, and the message will be formatted to the screen width of each reader. If you wish to force a RETURN you must use a control A. This will drop the cursor to the next line as if you had entered a RETURN. Only this return will be saved with the message.

NOTE: please place a message in the base in which it belongs, i.e., GENERAL, ADS, ST (use the

Z command to switch bases before you use the E command). Personal messages should be left using the L command.

F — 8 bit download directory. This will first display a list of the download categories. Press a letter for the category you want. You will see a list of all the files your level will allow you to download. To select a file from this list, press a

D — for download, and then enter the number at the farthest left of the filename. Missing numbers cannot be downloaded unless your level is higher (by joining ACE)

G — log-off (when you're done). PLEASE LOG OFF BEFORE YOU HANG UP! Hanging up without logging off can cause numerous problems - get in the habit - it only takes a couple of seconds!

H — read this description.

I — ST downloads. Same as

F — described above. All files in the categories which are described as "SQUEEZED" must be unsqueezed after you download them before you can use them. The unsqueeze program is in the first "SQUEEZED FILE" category.

L — Leave a message for a specific person. The person will be shown the message the next time they log on.

M — This allows you to send a message to the SYSOP via the printer. Use this for password applications and any messages the SYSOP will need to keep.

Q — If you know the subdirectory and the filename that you want to download, then you may use this command. When it asks you for the filename, enter the subdirectory

filename - note: no extension. Example:

GAME

CRIBBAGE

R — Read a message(s). You may enter a single number, a +

for the first message, a -

for the last message, or press

RETURN

to start from the last message (same as -). Use

N — to read the next message, or

P — to read the previous message.

N — will return to the main menu if there are no more messages. When reading a message you may immediately stop and go on to the next message by pressing any key. You will continue in the direction you were last going.

S — Tells you your current status of download level, online time, etc.

T — Allows you to select the screen width of your display. This will affect the word wrap on all text except the menus (80 column users will have a different function menu).

U — Upload a file. You will first be informed of how many double density sectors are free - if your program won't fit, then please don't try to upload it! You will next be asked for the filename. If the file already exists, you will be informed as such, and must use a different

filename. You will then be asked to describe the file. Use the same editing commands as the normal message editor. Once saved, you will be prompted to begin uploading.

V — Read the system help files - there are also occasional reviews, benchmarks and miscellaneous reading materials here.

X — Xpert mode - gives you 3 options which are saved and used whenever you log on:

Short prompts:

Y — if you are VERY familiar with the BBS.

Clear screen:

Y — if you want your screen cleared before menus, msgs, etc.

Continuous read:

Y — allows you to scroll through messages without pressing N for next. Press any key to stop. At 1200 baud, it'll be too fast to read, use it only if you are capturing all messages.

Y — Yell for SYSOP - sets a flag that you want to type to the SYSOP. If the SYSOP sees it before you log off, then he will type to you.

Z — change to a different message base. There are currently 4 bases:

1-GENERAL

2-ADS

3-ST

4-C

* — If you have a password, then the highest message number in every base will be saved with your password every time you log off. When you press *

from the main menu, you will be given a list of all message bases containing new messages since the last time you logged on. Select the base you want to read, and it will re-read the last message you read there. As soon as you recognize the message, hit the space bar or some other key to go on to the next message. If you don't recognize it, then the base has been compacted, and you should use the

P — key to back up till you see a message you recognize. Once you have read all messages in that base, you will again be given a list of bases with new messages. Enter a base number, or press RETURN to exit to the main menu.

? — prints the main menu.

0-9 — if you're in Atari ATASCII translation, and your screen is set to 40 columns, then you can view "movies" by pressing any number for 0 through 9. Pressing any key while watching a movie will abort it and return you to the main menu.

½ — This lists the BBS stats - the first 6 lines contain an * for every message, and a : for every deleted message. The bottom 2 lines displays the number of callers for the month, how many of those were 16 bit users and how many were 8 bit users; how many callers were at 300 baud, and how many of the callers were guests rather than password holders.

NOTE: 300 baud users - you can use control C to ignore the log-on logo, the bulletin, the menu, and most other lengthy prompts. 1200/2400 users you can do this too if you're fast eno

DOWNLOAD LEVELS:

2-ACE guest

5-ACE member

From Saturday morning until late Saturday night, the board will be closed to all but ACE members. This is to allow ACE members easier access. If you want higher access, why not join ACE? After all, it's through subscriptions that this board is possible.

USER HINTS

Tech Tip of the Month from JACS

(reprint: June, 1986 SNACC)

The ST monitor jack has a composite video output on it. This means it can be connected to a standard video monitor, with less resolution. The tip is to add a RCA phono jack to the back of the machine, rather than trying to find the multi-pin DIN plug Atari uses. This plug is next to impossible to find. Consult your ST Operation Manual for the exact pin-out for this connection.

Ever want to drag a file out of a desktop window without activating that window (and then having to re-activate the window you really want)? Well, a user seems to have found something else out about the ST which neither Atari nor DRI seems to have noticed. If you hold down the right mouse button, you can perform file functions on an inactive window with the left button, just as if it was an active window. You can even copy a file between two inactive windows.

BUMPAS REVIEWS

PRINT MASTER

Print Master (\$14, Unison World, Inc., 2150 Shattuck Ave., suite 902, Berkeley, CA 94704) turns your ST and a printer into a nifty little graphics machine. It's easy to create and produce signs, calendars, banners and stationery letterheads. Menus provide you with selections which place your choice of 122 pre-designed graphics, 11 border designs and 8 type fonts onto the product you're designing. A utility is provided to create your own graphics designs. The text editor permits you to align the text to the right, center or left. Changing fonts in a line, as well as changing the size and appearance of the font is supported. There is a graphics editor for creating your own graphics, but I was unable to get this to work properly. It kept drawing in white on a white background (monochrome). I had to fill with black, and use the erase function to draw in white. Another minor problem is in the Stationery mode. When I used the option to print lines at the bottom of the page, the last line always was off the paper even when I started the top at the lowest possible point. You might want to use 14" paper, or take the printer off-line and roll the platen back a couple of lines.

You can buy other products to enhance your use of this program. They have "Art Gallery" disks which contain additional items to use in your designs. A book, "The Creative PrintMaster", contains design tips. You can order colored printing paper, ribbons, T-shirt transfers.

The 34-page manual is well-written. But the program is so good, you'll probably not find the need to refer to the documentation very often. You can boot this one up and step right into printing useful and attractive graphics. The only configuring required is to select a printer driver from several pages of choices. Your printer is probably one of the listed ones. I use a Star SG-10 in IBM mode. So I selected the IBM printer driver.

HELPMASTER ST

Royal Software (2160 West 11th Ave., Eugene, OR 97402) makes this desktop accessory available for \$40. It contains a calendar with appointment alarms, calculator and phone pad.

The calculator provides you with an option to save the results of calculations, and to use them in an application file. For instance, while writing a letter with 1st Word, you want to calculate some number. You can pull down the calculator and try to remember the result as you type it into your text. Or, you can save the result, and then "Read" the calculation file into your text. If you're using ST Writer you'll have to "Merge" the file.

The phone pad can contain up to 100 telephone numbers. If you're connected to a modem, you can dial any selected phone number. Be sure to pick up the telephone receiver. Frequent modem users might forget this once in awhile. The nice thing about this is you don't have to remember any phone numbers. Just look them up and click on "Dial". The appointment calendar is the best feature of this package. Up to six alarms can be set for each day. When an alarm goes off, a bell will ring and a dialog window will appear inside any GEM application to show you the alarm. I find this a very useful accessory, and I've been using it every day. I've never needed more than 6 alarms in one day, but I suppose it's possible someone might. I also think the alarm messages might contain more information. Only one short line of information is permitted to describe the alarm message. I believe the program might be improved if the display of any month's calendar showed the days upon which alarms are scheduled.

I found a couple of bugs, which I understand are being corrected now. Sometimes when I try to drag a file into the trash can, the system bombs out in one of two ways. Sometimes it locks up with multiple bombs; sometimes two bombs just dump it to the ROM desktop in low resolution. I've found deleted files reappear on the directory after putting them in the trash can, too. This can be disconcerting. On the other hand, the one or two shortcomings of the package are balanced by the fact that these accessories take up very little space in memory. If the alarm messages could be longer, this might take up memory you'll want elsewhere. I find I use these accessories more than any other I've yet seen. If you schedule a lot of appointments and make a lot of phone calls, you'll find this package helpful, too.

U.S.A.A.F.

The American WW2 daylight bombing campaign is brought to your 8-bit Atari by SSI (\$60). The Atari can play either side, and a two-player option is available. You can play the entire campaign (each turn is one day) of more than 700 turns. Or you can select one of two shorter campaigns, or a "Short" game of 30 turns (one month). You

can handicap either side for play balance.

The goal of the U.S. player is to bomb the economy of Europe into rubble. The German player has the job to try to prevent this. The map covers England, France, the low countries, Italy, Germany, Luxembourg, Switzerland, Denmark, Austria, Czechoslovakia, Hungary, Poland, Rumania, Yugoslavia, and Bulgaria.

The U.S. player launches bombing, escort and fighter sweep missions from one of four bases in England, Tunisia, Libya or Italy. The German player has 47 airfields from which to launch intercept or patrol missions. These airfields may be relocated to any one of approximately 50 additional sites which may contain an airfield. Any aircraft entering Swiss airspace is attacked by Switzerland.

There are about 15 different target types to bomb. Germany has 13 aircraft types, the U.S., 11. Reinforcements and replacement aircraft appear automatically. The German player can control production to some extent. But be warned: The documentation covering production does not seem to match the play action in the game. The documentation says certain changes in production only reduce production in that city by 20%. I find ANY change I make reduces production by 60%, except for producing additional aircraft types at one city. You can produce up to 3 different types of aircraft in each city. If you boost the number of factories producing aircraft which can't yet appear, you may get them "early". For instance, the Me262A is scheduled to appear in June, 1944. It's marked with an "*" to say it might be "late". Where the manual describes adding factories to increase the odds the plane will appear "early" says the latest date it will appear is November, 1944. Apparently, the documentation indicates it can't appear earlier than June. But if you get the "die-roll" each month beginning in August, 1943, it seems it should appear in March, 1944. In fact, I began getting the jets in April, 1944.

Once the U.S. player sets up the attacks, he just watches the results. The German player can move flak around, re-locate airfields and shuffle interceptor groups around. He can stop the action at any time to assign interception and patrol missions. This is probably the most disk-intensive game I've ever seen. There are a minimum of 4 disk reads EACH turn. I advise you to make a scratch copy of the back side of the disk and use that instead of the SSI disk, in case you wear it out! Some turns might require 12 (or more!) disk reads. If ever a game called for ramdisk support, this one does. And with more people using 130 XEs and expanded memory 800s and 128s, I don't think it's unreasonable to expect SSI to keep up with what's happening and offer this support. A version of this game which supports ramdisks might improve play time by a factor of 10 or more.

At the end of each turn the players can check their status. The numbers of aircraft lost and target damage is displayed. One can check the number of serviceable and unserviceable aircraft and crew morale and expertise of their aircraft groups. A P40 or Me109 group with high expertise and morale will outfight an enemy with Fw190s or P51s. Attrition favors the U.S. The U.S. receives more than 50 replacement aircraft a turn, with reinforcements above that. The German player can only receive that many if all production is turned over to some type which becomes obsolete, like the Me109G. If you want to produce advanced aircraft, you're going to fall way behind in the numbers game. Additionally, attacks on airfields, aircraft factories and fuel sources can reduce aircraft availability even further. The German player can affect U.S. aircraft availability only by shooting them down.

Recalling a saved game is more of a problem than in previous SSI games. First, you must remember to reset the options menu each time. Otherwise, each time you recall the game in solitaire play, you'll be in the opposite side's mode. If you can resist the temptation not to look and/or modify your enemy's turn, you can "Quit" out of the mode and get back to your own. Also, it seems if you save the game at any opportunity other than just after the "Overnight Activity" phase, upon recall the sides will be switched. I'm told SSI is correcting this problem, which doesn't affect two-player games.

This is a great game which can be made much greater with ramdisk support. Some players might tire of the disk read every time the German player decides to send up another one of his dozens of air groups.

ABACUS

The latest ST books from Abacus are here: **ATARI ST Graphics and Sound** and **ATARI ST Basic Training Guide**. These books are both more than 300 pages, fully indexed and with complete tables of contents. They contain extensive examples, problems and appendices to help you to understand the well-designed topics. At \$20 each, these books are welcome and helpful additions to the fine library of books to help us understand the ST which is coming from this German company.

MOVIE

BY RALPH WALDEN

```

/* Movie Maker 6/86 by Ralph Walden
 */

#define START 1
#define SELECT 2
#define OPTION 3

char *buf, /* pointer to buffer area */
    *ptop, /* permanent top of buffer
*/
    *tptr, /* temporary pointer */
    *bptr, /* points to last char in b
uffer */
c, /* command from user */
fname[64]; /* filename (allow room
for Sparta subdirectories) */

int maxlen, /* size of buffer */
delay; /* determines baud rate */

Main() {
    int iocb;
    bptr=buf=lmem(100); /* setup buff
er pointer */
    ptop=highmem();
    maxlen=ptop-bptr; /* size of buffer
*/
    setblock(buf,maxlen,' '); /* set b
uffer to spaces */
    poke(0x52,0); /* left margin at ze
ro */
    while(1) { /* loop forever */
        c=menu(); /* display menu and ge
t command */
        switch(toupper(c & 0x7F)) {
upper case, no inverse
            case 'L':
                printf("Filename to load:");
                if(gets(fname)>0) {
                    if(!iocb=openfile(fname,'r
','SUP')) break;
                    bptr=buf+bget(buf,maxlen,
iocb);
                    close(iocb);
                }
                break;
            case 'A':
                printf("Filename to apend:");
                if(gets(fname)>0) {
                    if(!iocb=openfile(fname,'r
','SUP')) break;
                    bptr=bptr+bget(bptr,maxle
n,iocb);
                    close(iocb);
                }
        }
    }
}

```

```

        break;
        case 'S':
            printf("Filename to Save to:
");
            if(gets(fname)>0) {
                if(!iocb=openfile(fname,'w
','SUP')) break;
                bput(buf,bptr-buf,iocb);
                close(iocb);
            }
            break;
        case 'D':
            printf("Directory filespec:");
        ;
            if(gets(fname)>0) {
                if(!iocb=openfile(fname,'d
','SUP')) break;
                while(cgets(fname,iocb)>0)
cput(buf,0);
                close(iocb);
                user(); /* wait for keypre
ss or function key */
            }
            break;
        case 'E':
            delay=0; /* full speed */
            view(); /* for viewing */
            edit();
            poke(712,0); /* restore bord
er to black */
            break;
        case 'C':
            bptr=buf;
            putchar('\f'); /* clear scre
en */
            edit();
            poke(712,0); /* restore bord
er to black */
            break;
        case 'V':
            printf("\n0-full speed ahead
!\n1-1200 baud\n2-300 baud (yuck!)\n
");
            do c=getkey();
            while(c < '0' || c > '2') / *
wait for correct response */
            delay=c-'0';
            view();
            if(!delay) user(); /* view()
only pauses at slower speeds */
            break;
        case 'Q':
            exit(); /* back to DOS */
            break;
        default:
            break; /* ignore it, back to
top of loop */
        }
    }

menu() {
    printf("\f%D bytes free.\n\n",ptop
-bptr);
    printf("D - directory\n");
    printf("L - load file\n");
    printf("S - save file\n");
    printf("A - append to current movi
e\n");
    printf("V - view\n");
    printf("E - edit\n");
    printf("C - create new movie\n");
    printf("Q - quit\n");
    printf("\nEditing commands:\n\nSTA
RT - backup one\nSELECT - exit to me
nu\n");
    printf("OPTION - forward one\n\n");
    return getkey();
}

view() {
    putchar('\f');
    tptr=buf;
    if(delay == 2) { /* 300 baud */
        while(tptr<bptr) {
            putchar(*tptr++);
            clrtime();
            while(gttime() < delay); /* wai
t */
        }
    }
    else if(delay == 1) { /* 1200 baud */
        while(tptr<bptr) {
            putchar(*tptr++);
            if(tptr<bptr) putchar(*tptr++);
            clrtime();
            while(gttime() < delay); /* wai
t */
        }
    }
}

edit() {
    int i;
    poke(712,0xF4); /* bright border */
    while(1) {
        while(!console() && bptr<ptop)
if(inkey()) *bptr++=getchar();
        if(console() == START && bptr>bu

```

MOVIE CONTINUED

```

f) ${
    --bptr;
    c=&bptr & 0x7F; /* ignore inverse */
    if((c<27 || c>31 && c<123) &
    & peek(0x55)) putchar('\b'); /* if possible, just backspace */
    else ${*
        switch(c) ${/* reverse curs or movements */
            case '\l':
                putchar('\r');
                break;
            case '\r':
                putchar('\l');
                break;
            case '\u':
                putchar('\d');
                break;
            case '\d':
                putchar('\u');
                break;
            default: /* if all else fails, show it from the beginning */
                delay=0;
                view();
                break;
        }
    }
    wait(); /* pause between console key movements */
}
else if(console() == OPTION) ${
    for(i=0; console() == OPTION & & i<10 && bptr<pstop; ++i) ${/* 10 times at normal speed */
        putchar(*bptr++);
        wait();
    }
    while(console() == OPTION && bptr<pstop) putchar(*bptr++); /* after 10 times at normal, full speed */
}
else if(console() == SELECT) return;
}
wait() ${*
    clrtime();
    While(gtime() < 8);
}

user() ${/* wait for keypress or console key */}
    while(!console() && !inkey());
    poke(764,0xFF); /* ignore any key press */
}

```

```

$)
opnfile(file,mode,ext) /* returns iocb, or zero if can't open */
char *file,mode,*ext;
${*
    int iocb;
    normalize(fname,"SUP"); /* add drive and extension if not specified */
}
if((iocb=copen(fname,mode))<1) ${*
    printf("Can't open %s\nPress key to continue...\n",fname);
    getkey();
    return 0;
}
else return iocb;
}

```



SECRET CODES

(Reprint: June, 1986 OrnJuice)

Those of us who use SynFile+ and SynCalc with our 8-bit Ataris may be interested to know we can use both expanded and condensed print with our dot matrix printers.

Hidden on one of the last pages of the SynCalc manual is a short explanation of how to engage condensed print mode for some printers. I'll use Star SG printer series codes for examples. To engage condensed print mode, I press CTRL-O. The manual doesn't explain how to get back to the standard print mode after engaging condensed print. This started my investigation into the **CONTROL CODE CAPER**.

First I implemented the above example on a test SynCalc spreadsheet. CTRL-O puts a small solid block on screen. This looks familiar. Researching my files, through old programming books, digging in my closet through old game programs, I finally came to the bottom drawer of my computer cabinet deep below the game paddles and loose printer paper. The answer! A table of ATASCII characters on page 41 of "Understanding Atari Graphics" by Michael Boom shows CTRL-O, decimal 15, produces a solid block. Those of us who own Epson or compatible printers should know decimal code 15 engages condensed print. So I went back to the printer manual to see how many other decimal codes are available using this method.

Some printer codes are a single digit, some others are two digits, and others still require multiple commands to enter different print modes. I find a decimal 15 engages condensed print; decimal 18 engages standard pica print. With further research, I discovered decimal 14 enters expanded print for one line, and decimal 7 rings the printer's bell. SynCalc does not accept all these different printer codes, so to simplify things I've included a small table of commands which both SynCalc and SynFile+ accept:

Keystroke	Decimal Code	Printer Function
CTRL-O	15	Condensed Print
CTRL-N	14	1 Line Expanded Print
CTRL-R	18	Standard Pica Print
CTRL-G	7	Ring Printer's Bell

All of these codes work with both SynFile+ and SynCalc. With SynFile+, you should enter the codes during data entry, not when creating data fields. Combining codes to create expanded condensed print also works, but keep in mind that condensed printing remains engaged even though the expanded print only continues to the end of the line. You must use CTRL-R to cancel condensed mode and return to standard pica print.

Yet another case of intrigue is solved. Good luck in your experiments!

— Randy Dorn

CHARLIE CHICKEN

```

; *****
; * CHARLIE CHICKEN ADVENTURE #1 *
; *****

BYTE RAMTOP=106,curpos,loseflg,
ccpos,ccspin,ccsit,sndflg,
sunpos=[48],timeflg,sndpos=[0],
spincnt,brkcnt,goodcnt,winflg,
ccx,ccy,CONSOL=53279,foxflg,
msgflg,snaflg,sunspeed,sitflg,
sitpos,snpos,taken,spinmax=[10]
CARD pbase
BYTE ARRAY twr{45},max{36},stat{36},
dir{36},pnt{36},sndray,cseti=
[0 3 7 7 15 15 15 15 0 192 224 22
4 240 240 240 240 15 15 15 7 7 3 3
1 240 240 240 224 224 192 192 128
0 0 7 15 31 63 127 127 0 0 128 19
2 224 224 240 240 127 127 63 63 31
15 7 0 240 240 240 224 224 192 128 0
1 3 3 7 7 15 15 15 128 192 192 224
224 240 240 15 15 15 15 7 7 3 0 2
40 240 240 224 224 192 0 0 0 1 3
7 7 15 15 0 0 224 240 240 252 254 2
54 15 15 15 7 7 3 1 0 254 254 252 2
52 240 240 224 0 0 12 24 24 62 56 56
62 0 240 120 120 28 124 124 252 30 3
1 15 15 7 7 3 1 240 120 176 112 224
224 192 128 0 3 5 130 193 97 113
63 0 192 160 65 131 134 142 252
63 63 63 31 31 7 0 0 252 252 252
248 248 224 0 0 0 1 32 17 8 5 2 8
4 0 0 132 8 144 32 64 42 40 2 5 8
17 32 1 0 20 64 32 144 8 132 0 0
255 255 255 255 255 255 255 255 255 1,
ey=[5 5 5 5 5 8 8 8 8 8 11 11 1
1 11 11 11 14 14 14 14 14 14 14 17
17 17 17 17 20 20 20 20 20 20 20 20
1, sun=[0 24 60 126 126 255 255 255 25
5 126 126 60 24 0], moon=[0 24 60 118 98 225 225 225 22
5 98 118 60 24 0], straw=[10 72 81 91 0 91 0 91 0 133
136 121 0 121 0 121 0 91 81 72
0 72 8 72 81 91 81 72 0 81 0 81
11, foxsnd=[14 162 0 136 0 108 0 81 0 4
2 102 14 0 7 108 0 121 0 136 0 144 0
11, losesnd=[40 243 243 10 243 40 243 2
04 10 217 40 217 10 243 40 24
3 10 255 60 243 11
PROC Vblank()
BYTE j
FOR j=0 TO 44 DO
IF twr(j)>0 THEN twr(j)==-1 FI 0
D
I $4C $62 $E4 J
RETURN
PROC Setvect=$E45C(BYTE areg,xreg,yreg)
PROC Snd()
BYTE sel,dur
IF sndflg>0 THEN
IF twr(38)=0 THEN
twr(38)=dur
sel=sndray(sndpos)
IF sel<61 THEN
IF sel=0 THEN SndRst()
ELSEIF sel=1 THEN
sndflg=0 SndRst() sndpos=0
ELSE dur=sel twr(38)=0 FI
FI
sndpos==+1
FI
IF twr(38)<12 THEN
Sound(0,sel,10,twr(38))
ELSE Sound(0,sel,10,10)
F)
RETURN
PROC Instr()
Graphics(0) Print("**** INSTRUCTION
;
```

BY STAN OCKERS

```

    FI
    FI
    RETURN

PROC Title()
CARD dlist,delay
Graphics(2) Poke(752,1) Poke(712,3
3)
dlist=PeekC($230) Poke(dlist+6,2)
Poke(dlist+11,6) Poke(710,160)
Position(7,1)
PrintD(6,"The Adventures Of
")
Position(2,4) PrintD(6,"CHARLIE CH
ICKEN")
Position(6,6) PrintD(6,"part i")
Position(3,8) PrintD(6,"the hatchi
ng")
PrintE("By Stan Ockers")
PutE()
Print("Written in Action!
")
FOR delay=1 TO 60000 DO OD
PutE() PutE() PutE()
PrintE("OPTION - Instruct
ions")
PrintE("START - Begin ga
me")
sndray=straw
sndflg=1 tmr(38)=8
DO
IF CONSOL=3 THEN Instr() FI
Snd()
IF sndflg=0 Then sndflg=1 FI
UNTIL CONSOL=6
OD
SndRst() sndflg=0 sndpos=0
RETURN

PROC Begin()
BYTE j
FOR j=0 TO 35 DO
    tmr(j)=0 max(j)=16 stat(j)=1
    OD
    tmr(36)=0 ccspin=0 ccsit=0 spincn
    t=0
    brkcnt=0 goodcnt=36 winflg=0
    loseflg=0 sndflg=0 msgflg=0 snaf
    g=0
    SetBlock(pmbase+1280,512,0)
    sitflg=0 foxflg=0 svpos=58 timewf
    l
    g=1
    sunpos=48 sunspeed=10 taken=0
    sndpos=0 SndRst()
RETURN

PROC Draw4(BYTE j,strt)
    Position(ex(j),ey(j))
    PutD(6,strt) PutD(6,strt+1)
    Position(ex(j),ey(j)+1)
    PutD(6,strt+2) PutD(6,strt+3)
    RETURN

PROC Egg(BYTE j,dir)
    BYTE ARRAY eggdir=[2 6 10 14]
    Draw4(j,eggdir(dir))
    RETURN

PROC Bikegg(BYTE j)
    Position(ex(j),ey(j))
    PutD(6,32) PutD(6,32)
    Position(ex(j),ey(j)+1)
    PutD(6,32) PutD(6,32)
    RETURN

PROC Spin()
BYTE j,r,HP0SP1=53249
CARD pos,fill
FOR j=0 TO 35 DO
    IF stat(j)=1 THEN
        IF tmr(j)=0 THEN
            IF Max(j)<spinmax THEN
                dir(j)=+1
                IF dir(j)=4 THEN dir(j)=8
            FI
            Egg(j,dir(j))
            IF j=ccpos THEN
                r=Rand(4) ccspin=1
                HP0SP1=ccx*24+68
                pos=pmbase+1356+24*ccy
                fill=$E200+8*(28+r)
                MoveBlock(pos,fill,8)
            FI
            IF Rand(3)=1 THEN
                max(j)=+1 FI
            IF max(j)=spinmax THEN
                IF dir(j)>pnt(j) THEN
                    max(j)=spinmax-1
                ELSE dir(j)=pnt(j) FI
            FI
            tmr(j)=max(j)
            IF Max(ccpos)=spinmax AND
                ccspin=1
                THEN loseflg=1 FI
            FI
            FI
        OD
    RETURN

PROC Startspin()
BYTE j
IF max(curopos)=spinmax THEN
    IF Strig(0)=0 AND timewflg=1 THEN
        Egg(j,dir(j))
    ELSEIF stat(j)=2 THEN Draw4(j,18
)
    FI
    FI
    RETURN

PROC Shuffle(BYTE nbr1,nbr2)
BYTE j,r
FOR j=0 TO 35 DO
    stat(j)=0 dir(j)=Rand(4) OD
FOR j=1 TO nbr1 DO
    DO r=Rand(36)
    UNTIL stat(r)=0 OD
    stat(r)=1
    OD
FOR j=1 TO nbr2 DO
    DO r=Rand(36)
    UNTIL stat(r)=0 OD
    stat(r)=2
    OD
RETURN

PROC Fillpnt()
BYTE x,y,j,pnt1,pnt2
DO ccpos=Rand(36)
UNTIL stat(ccpos)=1
OD
ccx=ccpos MOD 6 ccy=ccpos/6
FOR j=0 TO 35 DO
    x=j MOD 6 y=j/6
    IF x<ccx THEN pnt1=3
    ELSEIF x>ccx THEN pnt1=1
    ELSE pnt1=4
    FI
    IF y>ccy THEN pnt2=2
    ELSEIF y<ccy THEN pnt2=0
    ELSE pnt2=4
    FI
    IF j=ccpos THEN pnt(j)=Rand(4)
    ELSE
        IF Rand(2)=0 THEN
            IF pnt1<4 THEN pnt(j)=pnt1
            ELSE pnt(j)=pnt2 FI
        ELSE
            IF pnt2<4 THEN pnt(j)=pnt2
            ELSE pnt(j)=pnt1 FI
        FI
    FI
    OD
RETURN

PROC Draweggs()
BYTE j
FOR j=0 TO 35 DO
    IF stat(j)=1 THEN
        Egg(j,dir(j))
    ELSEIF stat(j)=2 THEN Draw4(j,18
)
    FI
    FI
    RETURN
}

```

CHARLIE CHICKEN

```

ELSEIF stat(j)=3 THEN Draw4(j,54)
)
ELSE Blkegg(j)
FI
OD
RETURN

PROC Msg1(BYTE ARRAY msg)
Position (9-msg(0)/2,1) PrintD(6,m
sg)
RETURN

PROC Msg2(BYTE ARRAY msg)
Position (9-msg(0)/2,2) PrintD(6,m
sg)
RETURN

PROC Msg3(BYTE ARRAY msg)
Position (9-msg(0)/2,3) PrintD(6,m
sg)
RETURN

PROC Msgclr()
BYTE ARRAY blinks=""
"
Position(1,1) PrintD(6,blinks)
Position(1,2) PrintD(6,blinks)
Position(1,3) PrintD(6,blinks)
RETURN

PROC Msgset(BYTE time)
tmr(39)=time msgflg=1
RETURN

PROC Msgdly()
IF msgflg=1 THEN
  IF tmr(39)=0 THEN
    msgflg=0 Msgclr() FI
  FI
RETURN

PROC Playfld()
BYTE j
Graphics(17) Poke(756, RAMTOP+1)
Draweggs()
FOR j=1 TO 18 DO
  position(j,0) PutD(6,62)
  Position(j,4) PutD(6,62)
  Position(j,23) PutD(6,62)
OD
FOR j=0 TO 23 DO
  Position(0,j) PutD(6,62)
  Position(19,j) PutD(6,62)
OD
Poke(559,62) Poke(53277,2)
Poke(788,56) Poke(711,72)
Poke(710,152)

Poke(712,32) Poke(709,8)
RETURN

PROC Foxset()
IF Rand(6)=0 THEN foxflg=1
  tmr(41)=155+Rand(100) FI
RETURN

PROC Crack(BYTE j)
BYTE k,r
IF stat(j)=1 THEN
  stat(j)=2 Msgclr()
  Msg2("CRACK") Msgset(60)
  FOR k=1 to 250 DO
    r=Rand(50) Sound(1,r+50,2,12)
    OD SndRst()
    brkcnt==+1 goodcnt==+1
    IF j=ccpos THEN loseflg=2 FI
  FI
RETURN

PROC Movesun()
BYTE bkgcol,eggcol,r
IF tmr(36)=0 THEN
  tmr(36)=sunspeed sunpos==+1
  Poke(53250,sunpos)
  IF sunpos MOD 5 =0 THEN
    bkgcol=Peek(712) eggcol=Peek(7
09)
    IF timeflg=1 THEN
      IF sunpos<100 THEN
        IF bkgcol<35 THEN
          Poke(712,bkgcol+1)
          Poke(709,eggcol+4) FI
        ELSEIF sunpos>175 THEN
          IF bkgcol>32 THEN
            Poke(712,bkgcol-1)
            Poke(709,eggcol-4) FI
        FI
      FI
      IF (sunpos MOD 3)=0 THEN
        IF sunpos<124 THEN svpos==+1
        ELSE svpos==+1 FI
        IF timeflg=1 THEN
          MoveBlock(pmbase+1536+svpos,5
un,14)
        ELSE
          MoveBlock(pmbase+1536+svpos,m
oon,14)
        FI
        IF sunpos=200 THEN sunpos=48
        IF timeflg=0 THEN timeflg=1
          Draweggs() sunspeed=10
          IF ccsit=1 THEN winflg=1 FI
        FI
        tmr(40)=155+Rand(100)
      FI
    FI
  RETURN

PROC Snatchi()
BYTE j,p
CARD K
IF snaflg=1 THEN
  IF tmr(40)=0 THEN snaflg=0
  DO j=Rand(36)
  UNTIL stat(j)=1
  OD
  Msg2("ONE") Msg2("egg McMuffin
!")
  Msg3("PLEASE")
  FOR k=1 TO 400 DO
    Movesun() Spint()
  OD Draw4(j,26)
  snaflg=1 Poke(712,32)
ELSE timeflg=0 Poke(712,0)
  foxset() sunspeed=5
  IF sitflg=1 THEN
    blkegg(sitpos) FI
  IF Rand(3)=0 THEN
    r=Rand(36) Crack(r) FI
  FI
  FI
RETURN

PROC Cursor()
BYTE stkdir,crow=[0],ccol=[0],
HP0SP0=53248,curspd=[10],j,r
BYTE ARRAY curs=[24 24 255 24 24]
IF tmr(37)=0 THEN
  stkdir=Stick(0)
  SetBlock(pmbase+1110+24*crow,5,0
)
  IF (stkdir=14) AND (crow>0) THEN
    crow==+1
  ELSEIF (stkdir=13) AND (crow<5)
  THEN
    crow==+1
  ELSEIF (stkdir=11) AND (ccol>8)
  THEN
    ccol==+1
  ELSEIF (stkdir=7) AND (ccol<5) T
HEN
    ccol==+1
  FI
  curspos=6*crow+ccol
  IF timeflg=0 AND stkdir<>15 THEN
    Crack(curspos) FI
    MoveBlock(pmbase+1110+24*crow,cu
rs,5)
    HP0SP0=68+24*ccol tmr(37)=curspd
  FI
  RETURN

PROC Snatchi()
BYTE j,p
CARD K
IF snaflg=1 THEN
  IF tmr(40)=0 THEN snaflg=0
  DO j=Rand(36)
  UNTIL stat(j)=1
  OD
  Msg2("ONE") Msg2("egg McMuffin
!")
  Msg3("PLEASE")
  FOR k=1 TO 400 DO
    Movesun() Spint()
  OD Draw4(j,26)

```

CHARLIE CHICKEN CONTINUED

```

FOR p=1 TO 25 DO
    Sound(1,75-p,10,12)
    Movesun() Spin()
    Movesun() Spin()
OD
SndRst() Msgset(100)
Blkegg(j) goodcnt=-1 stat(j)=
taken+=1
IF j==ccpos THEN loseflg=3 FI
FI
FI
RETURN

PROC Fox()
BYTE row,col
IF foxflg=1 THEN
    IF tmr(41)=0 THEN foxflg=0
    Msg1("FOX") Msg2("IN THE")
    Msg3("HENHOUSE!") sndray=foxsn
    d
    Msgset(200) sndflg=1
    WHILE sunpos<150 DO
        Row=5+Rand(15) col=1+Rand(15
    )
        Position(col,row) PrintD(6,"
    ")
        tmr(42)=30+Rand(100)
        DO Movesun() Snd()
        UNTIL tmr(42)=0
        OD Position(col,row)
        PrintD(6," ")
        tmr(42)=30+Rand(100)
        DO Movesun() Snd()
        UNTIL tmr(42)=0 OD
        OD
        Shuffle(goodcnt,brkcnt)
        Fillpnt() Draweggs() sitflg=0
    FI
FI
RETURN

PROC Hensit()
BYTE savest,key
key=Peek(764)
IF key=33 AND timeflg=1 THEN
    IF sitflg=0 THEN sitflg=1
    savest=stat(curpos) stat(curpo
s)=3
    Draw4(curpos,54) sitpos=curpos
    IF savest=2 THEN
        Msgclr() Msg2("YUCH!")
        Sound(1,111,12,14) tmr(43)=4
        DO Spin() Movesun()
        UNTIL tmr(43)=0 OD
        SndRst() Msgclr() FI
I
IF curpos==ccpos THEN csit=1 F
    Poke(764,255)
ELSE
    sitflg=0 stat(sitpos)=savest c
csit=0
    IF savest=1 THEN
        Egg(sitpos,dir(sitpos))
    ELSEIF savest=2 THEN
        Draw4(sitpos,18)
    ELSE Blkegg(sitpos) FI
    Poke(764,255)
FI
FI
RETURN

PROC Lose()
IF loseflg>0 THEN sndray=losesnd
    sndflg=1 Msgclr() FI
IF loseflg=1 THEN
    Msg1("YOU") Msg2("SCRAMBLED")
    Msg3("CHARLIE!")
ELSEIF loseflg=2 THEN Msg1("Charl
ie")
    Msg2("got") Msg3("cracked!")
ELSEIF Loseflg=3 THEN
    Msg2("CHARLIE") Msg3("GOT TAKEN
!")
FI
Msgset(255)
DO
    Msgdly() Snd()
    IF msgflg=0 THEN
        Msg2("press START") FI
    UNTIL CONSOL=6
OD
RETURN

PROC Score()
BYTE total
PutD(6,125) Poke(756,224)
PrintD(6,"*** final score ***")
PutDE(6)
PrintD(6,"starting pts 100")
PutDE(6)
PrintD(6,"-2 * SPINS --")
PrintBD(6,spincnt*2) PutDE(6)
PrintD(6,"-3 * CRACKED --")
PrintBD(6,brkcnt*3) PutDE(6)
PrintD(6,"-3 * TAKEN --")
PrintBD(6,taken*3) PutDE(6)
total=100-2*spincnt-3*brkcnt-3*tak
en
PrintD(6,"FINAL SCORE ")
PrintBD(6,total) PutDE(6) PutDE(6)
PrintD(6," press start")
DO Snd() UNTIL CONSOL=6 OD
RETURN

PROC Win()
BYTE row,col,r
BYTE ARRAY hearts=[251 219 123 91]
Poke(789,14) FOR r=0 TO 35 DO
    Blkegg(r) OD
    Msg1("CHARLIE") Msg2(" ")
    Msg3("hatched!") tmr(42)=255
    sndray=straw sndflg=1
DO
    row=5+Rand(17) col=1+Rand(18)
    Position(col,row) r=Rand(10)
    IF r<4 THEN PutD(6,hearts(r))
    ELSE PutD(6,32) FI
    IF tmr(42)=1 THEN
        Msg1("OPTION for score")
        Msg2(" ")
        Msg3("START to play")
    FI
    IF sndflg=0 THEN sndflg=1 FI
    Snd()
    UNTIL CONSOL=6 OR CONSOL=3
OD
IF CONSOL=3 THEN Score() FI
RETURN

PROC Main()
Init() Title()
DO Begin()
    Shuffle(goodcnt,brkcnt) Fillpnt()
    Playfld() tmr(37)=10
    DO
        Cursor() Spin() Movesun() Hensit
    ()
        Startspin()
        Snatch1() Msgdly() Fox()
        UNTIL loseflg=0 OR winflg>0
        OD
        IF loseflg>0 THEN Lose()
        ELSE Win() FI
    OD
RETURN

```



ROM I/O

An Input/OUTPUT Port for the Cartridge Slot of the 520ST
by Douglas P. B. Renaux — Feb. 27, 1986.

This document describes the hardware and software of a 8 lines input M16 lines output port. The circuitry is connected to the Cartridge Slot of the ATARI 520 ST. **IMPORTANT!!!** — INFORMATION IN THIS DOCUMENT CAN BE FREELY USED AND COPIED. HOWEVER, NO CHANGES CAN BE MADE TO IT WITHOUT CONSULTING THE AUTHOR FIRST, NOR THE AUTHOR'S NAME OMITTED IN THE COPIES.

Foreword

Although the Cartridge Slot is intended for data input, specifically from a ROM, it can be used as an output port. To substitute the READ/WRITE line, which is not available on the slot, I used a WRITE ENABLE LATCH. This latch is set by READING a memory location. The consequence of setting this latch is that the output latches will read the data bus during the next instruction, which will be a memory WRITE which outputs the desired data.

The signals available on the Cartridge Slot: D0..D15 - DATA BUS; A1..A15. The 15 least significant addresses from the ADDRESS BUS 3 SELECT - Active whenever a READ instruction accesses \$FBO to \$FBF \$UDS'LDS' - Upper and Lower Data Strobe.

The I/O PORT

The block diagram of the I/O PORT is in the file BLOCKDIA.PI3. Circuit diagrams are in ADDRESS.PI3 and IN/OUT.PI3. If you don't have these drawings then in appendix A you can read a description of the circuit.

The circuit of the I/O PORT is standard. An address decoder enables the output latches or the input buffers when the correct address is accessed. However, the output latches are only enabled if previously the WRITE ENABLE LATCH was set.

Address Decoder:

The lines A3..A15 are connected to NAND gates which detect when all these lines are at high level. The lines A2 and A1, select one of the four outputs of a decoder. The output of the NOR gate "a" is active when we read from \$FBFFF. To decode a read to this address we use the outputs of the two NAND gates, the output three of the decoder (active when A1 and A2 are high) and the line ROM 3 SELECT. The output of NOR gate "a" sets the WRITE ENABLE LATCH. The output of NOR gate "b" decodes the address \$XXFFC. To decode an access to these addresses (read or write), we use the outputs of the NAND gates, output two of the decoder and the line LDS'. ROM 3 SELECT cannot be used here because it is only active on read. When a write instruction accesses the address \$10FFFC (non-existent in 1MByte ST's) then the output of NOR gate "b" becomes active.

An AND gate enables the output latches to read the data bus if the WRITE ENABLE LATCH is set AND the output of NOR gate "b" is active. The output of NOR gate "b" is delayed (by four inverters) and then resets the WRITE ENABLE LATCH, just after the latches have read the data bus.

NOR gate "c" decodes a read to address \$FBFFFA, as gate "a" does, but using output one of the decoder. The output of gate "c" enables the buffers to copy their inputs to the data bus.

Components

- 2- 4068 NAND gates for address decoding
- 2- 4002 NOR gates for address decoding
- 1- 4556 1 OF 4 DECODER for address decoding
- 1- 4043 SR FLIP-FLOP write enable latch
- 1- 4049 INVERTER
- 1- 4081 AND 2- 40373 LATCHES (output latches)
- 1- 74LS244 BUFFERS (input buffers)

Expansion

This I/O PORT is easily expandable to as many I/O lines you want. Only one WRITE ENABLE LATCH is necessary if the expansion is made in a way that many write to port instruction resets the latch. Another solution is to have a reset latch address (as we have a set latch address). In this case the write to port instructions don't need to reset the latch, and you can output a block of data to several addresses without needing to set the latch in between. Only at the end of the block transfer will this be necessary. You can have another 8 inputs by just connecting another 74LS244 to D8..D15 of the data bus.

Software

To write to the output lines the contents of D0:

$\frac{1}{2} \& MOVE.W \frac{1}{2} \$FBFFF, D1 \frac{1}{2};$ D1 is destroyed, latch is set
 $\frac{1}{2} \& MOVE.W \frac{1}{2} D0, \$10FFFC \frac{1}{2};$ D0 is latched

To read the input lines to D0:

$\frac{1}{2} \& MOVE.W \frac{1}{2} \$FBFFFA, D0 \frac{1}{2}$

Connection to the ST

I could not find a experimenter PC Board with an edge connector to fit to the ST. So I made a convertor from the 40 pin edge connector (with spacing of 2mm) to a 64 pins euroconnector (DIN 41612). This convertor is simply a double sided PCB with 20 lines (on each side) of 1mm spaced by 2mm (center of one line to center of next line). These lines widen to 2.54 mm to fit the DIN connector. Line one of the ST goes to line 4 of the DIN connector. The tabs of the DIN connector are on the lower side.

I intend to connect some of the free lines of this connector to internal signals of the ST. Would be nice to have a standard connector between ST users. Whoever has ideas about which signals to connect and to which pin should let the others know about it.

Conclusion

It is possible to open the ST !! And this opens a lot of possibilities in hardware design. The 68000's power can be used outside the ST's box. But a complex machine like the ST cannot be easily expanded without having information about it. The information ATARI delivers only to software houses. Please ATARI, let everybody know about the ST. Let the POWER be available to all users.

— Douglas P. B. Renaux
Outshoornstraat 55
05622GX Eindhoven - NL
Tel 31-40-436930

[There was an Appendix to this article, but it was so scrambled on the BBS I got it from, it wasn't useful.

BRIMSTONE

This is the first time I have had the pleasure of reviewing an electronic novel. When I first started, I was not sure of the difference between the novel and an adventure game — I soon found out and was pleasantly surprised. An electronic novel is a book and an adventure game combined. To solve the puzzle, you must read the script and put what you have read to use.

The novel I am working on is "Brimstone", A Synapse & Broderbund Production written by Paul Author and programed by David Bunch, William Mataga and Bill Darrah. The script is extremely well written and contains some historical facts about a knight's apparel. The programming forces the player to ask and answer questions in full sentences. I find this immensely satisfying and it could be used as an educational program to help youngsters learn to read, think intuitively, reason and construct proper sentences. In my humble opinion, if the rest of the electronic novels this company puts out are as well written as this one they can't help but make money.

BRIMSTONE takes you, a gallant knight, on a quest through a terrifying under ground world called Ulro. You must first learn how to get into the glass case containing the knights armor (you arrive dressed in your night shirt), put it on, and then find out what in the world you are doing here. My family and I have spent many hours on this game so far and have yet to solve the puzzle — but at least we know what we're looking for now. Before we received this copy, it was almost impossible to get Sharon, my wife, involved in an adventure game. She plays this one with great enthusiasm. That in itself is a great tribute.

This adventure has a unique type of copy protection. As you enter the game it asks you to turn to a specific page line and word and enter it on the keyboard. This may cut down on the amount of pirating. If you are hoping the book itself will give hints about which moves to make you are sadly mistaken. You have to get into the game and ferret the information out for yourself. Luckily it is a lot of fun to do it.

I look forward to reviewing more of these electronic novels, Mr. Bumpas if you're reading this column that is a direct hint!!!

— Mike Rogers



MERLIN'S MAGIC SQUARE

MERLIN'S MAGIC SQUARE

Steve Hutton

Merlin's Magic square is an entertaining little program to test your logic skills. Instructions for play are included on the screen.

The program listing contains plenty of REMarks for you to follow. Pay particular attention to the inverse spaces required in lines 350 to 370. If you want to change the colour of the screen, alter the SETCOLOR statement in line 30. Also adding SETCOLOR 1,0,14 will brighten up the display.

```

10 REM *****
11 REM * MERLIN'S MAGIC SQUARE : *
12 REM * BY STEVE HUTTON : *
13 REM * 13/1/83 : *
14 REM *****
15 POKE 764,255
28 REM *** CLEAR SCREEN AND INHIBIT THE CURSER
30 ? CHR$(125):POKE 752,1:SETCOLOR 2,2 ,1:CLR
40 REM *** PRINT TITLE, INSTRUCTIONS AND COUNTER
50 POSITION 8,3:? "MERLIN'S MAGIC SQUA RE"
60 POSITION 3,8:? "PRESS 1-9":POSITION 2,9:? "TO MAKE THE":POSITION 4,10:? "NUMBERS"
70 POSITION 4,11:? "LIGHT UP":POSITION 3,12:? "OR GO OUT"
80 POSITION 27,8:? "THE OBJECT":POSITION 26,9:? "OF THE GAME":POSITION 27,12:? "IS TO LIGHT"
90 POSITION 27,11:? "UP ALL THE":POSITION 27,12:? "NUMBERS IN"
100 POSITION 26,13:? "AS FEW MOVES":POSITION 26,14:? "AS POSSIBLE":POSITION 6,18:? "NUMBER OF MOVES TAKEN = "
110 REM *** SET UP RANDOM SQUARE
120 Z=0:DIM SQ(9)
130 FOR D=1 TO 9
140 Z=Z+1
150 IF Z=1 THEN A=15:B=7:C=8:D=9
160 IF Z=2 THEN A=18
580 NEXT W
590 FOR W=8 TO 14
600 FOR U=26 TO 37
610 POSITION U,W:? " "
620 NEXT U
630 NEXT W
640 RETURN
650 IF SQ(1)=1 AND SQ(2)=1 AND SQ(3)=1 AND SQ(4)=1 AND SQ(5)=1 THEN GOSUB 67
660 RETURN
670 IF SQ(6)=1 AND SQ(7)=1 AND SQ(8)=1 AND SQ(9)=1 THEN STOP :REM GOTO 685
680 RETURN
685 POSITION 30,18:? M10
690 FOR I=1 TO 4
700 FOR H=200 TO 150 STEP -1
710 SOUND 0,H,10,8
720 NEXT H
730 NEXT I
740 SOUND 0,H,10,8
750 GOSUB 548
760 POSITION 3,9:? "SUCCESS!"
770 POSITION 4,11:? "WHY NOT"
780 POSITION 5,12:? "PLAY"
790 POSITION 4,13:? "AGAIN ?"
800 POSITION 27,9:? "PRESS ANY"
810 POSITION 28,10:? "KEY TO"
820 POSITION 28,10:? "KEY TO"
830 POSITION 28,11:? "START A"
840 POSITION 27,12:? "NEW GAME."
850 P=PEEK(764):IF P=255 THEN 850
860 IF P<>255 THEN GOTO 15
996 REM *** KEY NUMBER 1 PRESSED
997 GOSUB 2000
998 GOSUB 4000
999 GOSUB 5000
1000 SOUND 0,162,10,8
1010 A=15:B=7:C=8:D=9:Z=1
1020 IF SQ(1)=0 THEN GOSUB 350
1030 IF SQ(1)=1 THEN GOSUB 300
1040 IF SQ(1)=0 THEN SQ(1)=1:GOTO 1060
1050 IF SQ(1)=1 THEN SQ(1)=0
1060 SOUND 0,162,10,8
1070 RETURN
1097 REM *** KEY NUMBER 2 PRESSED
1098 GOSUB 1000
1099 GOSUB 3000
1100 SOUND 0,144,10,8
1101 A=18:B=7:C=8:D=9:Z=2
1102 IF SQ(2)=0 THEN GOSUB 350

```

FROM PAGE 6

```

170 IF Z=3 THEN A=21
180 IF Z=4 THEN A=15:B=10:C=11:D=12
190 IF Z=5 THEN A=18
200 IF Z=6 THEN A=21
210 IF Z=7 THEN A=15:B=13:C=14:D=15
220 IF Z=8 THEN A=18
230 IF Z=9 THEN A=21
240 Y=INT(RND(0)*2):SQ(0)=Y
250 IF Y=0 THEN GOSUB 300
260 IF Y=1 THEN GOSUB 350
270 NEXT Q
275 SOUND 0,60,10,8
280 GOTO 390
290 REM * SET UP END SQR(OFF=0)
300 POSITION A,B:? CHR$(137);CHR$(149)
;CHR$(143)
310 POSITION A,C:? CHR$(25);Z;CHR$(153)
)
320 POSITION A,D:? CHR$(139);CHR$(21);
CHR$(140)
330 RETURN
340 REM * SET UP RND SQR(OM=1)
350 POSITION A,B:? " ";REM INVERSE SPACES
360 POSITION A,C:? " ";CHR$(176+Z);"?";REM INVERSE SPACES
370 POSITION A,D:? " ";REM INVERSE SPACES
380 RETURN
382 GOSUB 650
385 REM *** THIS SECTION DEALS WITH THE KEYS 1-9
390 M1=31:M2=38:M3=26:M4=24:M5=29:M6=2
7:M7=51:M8=53:M9=48:M10=0
400 POSITION 30,18:? M10
410 P=PEEK(764):IF P=255 THEN 410
420 IF P=M1 THEN M10=M10+1:GOSUB 997
430 IF P=M2 THEN M10=M10+1:GOSUB 1998
440 IF P=M3 THEN M10=M10+1:GOSUB 2997
450 IF P=M4 THEN M10=M10+1:GOSUB 3998
460 IF P=M5 THEN M10=M10+1:GOSUB 4996
470 IF P=M6 THEN M10=M10+1:GOSUB 5998
480 IF P=M7 THEN M10=M10+1:GOSUB 6997
490 IF P=M8 THEN M10=M10+1:GOSUB 7998
500 IF P=M9 THEN M10=M10+1:GOSUB 8997
510 POKE 764,255:GOSUB 650
520 IF M10=3 THEN GOSUB 548
530 GOTO 400
540 FOR W=8 TO 12
550 FOR U=2 TO 12
560 POSITION U,W:? " "
570 NEXT U

```

IF YOU'VE GOT THE TIME

One excellent feature of modern computers is an internal clock. The internal clock allows automatic coding of data files and programs with the current time and date when they are saved. This means you can just look at the time stamp to find out which is the latest version of a file. It also allows you to utilize internal date functions in certain programs like VIP's @TODAY function. Some word processing programs allow pulling the internal date and time and placing them into your document. There are also many new "calendar", or reminder-type programs showing appointments when you reach the specified dates. Some even interrupt your current task to act as an alarm clock.

When I got my 520ST, one of the first things I did was set the time and date in the control panel. No more guessing which of the 33 versions of "LIFESAVE.C" is the most recent. To my surprise the date reverted to 05/29/85 every time I turned on the ST. This means I have to remember to reset the control panel every time the ST was turned on if I want to utilize the system time/date. Since remembering things like that are not my strong point, I spent a lot of time ending programs, correcting the time and date and re-entering the program. I also have quite a few files dated May of '85!

Mind Mine to the rescue! Mind Mine Computer Center (13256 N.E. 20th ST., Suite #4, Northrup Bldg., Bellevue, Wa 98005, (206)641-6138) has developed a battery powered internal clock to keep track of the time and date even if your ST is turned off. It's called the TIME-SAVER and comes with complete installation instructions. To install the TIME-SAVER, you open the case, unplug the keyboard, unplug an IC, plug in the TIME-SAVER, plug in the IC, plug in the keyboard and close up the case.

There! All done! What! You don't believe it's that simple? Well, OK! You're right! Those are the steps but there are a few "sticky" points you may run into. The instructions are broken down into twenty-two steps and include seven or eight warnings. It took me forty minutes to install the TIME-SAVER. I went extra-slow because I get very nervous pulling ICs and worrying about static discharges.

There are a few things I did differently than instructed. When pulling the keyboard plug, instead of pulling the wires, I used a long nosed pliers and levered up each end a bit, alternating until the plug came loose. If you try this, be very careful to not crush the plastic wire holder! Placing the IC into the TIME-SAVER board is the most difficult step. It is possible to bend the IC pins if the pressure isn't applied evenly, and it does take quite a bit of pressure to seat the IC. I placed the IC feet-up on the table and pressed the board onto it, but pushing the IC into the board may be just as easy. Either way, be careful. In two places, the components on the TIME-SAVER come very close to the shield or components on the computer. The instructions say if they come too close, stick some insulated tape between them. Since the tape is included, I feel it should be used as insurance even if the parts don't touch.

Lee Rahfeldt of MIND MINE told me some of the older STs have a capacitor which may be in the way of the TIME-SAVER board. If you have one of these, you will have to clip the lead to the capacitor, bend the capacitor out of the way and solder on a longer lead. My ST didn't have this problem so I don't know how much trouble it might be, but it doesn't sound too bad. Also, be warned: The screws from the back of the ST case may be slightly longer than the ones from the front. If you get them mixed up, you may end up redesigning your case!

If you are careful and follow instructions, you shouldn't have any trouble installing the TIME-SAVER but, If you have any doubts, have your dealer or a knowledgeable friend install it for a "nominal" price.

So! Does it work? Yep! Sure does! The instructions I have say to leave the computer on for 10-15 minutes to charge the battery. Lee told me to leave it on for eight hours the first time. After two hours, I turned the ST off for one minute and the time and date were still correct when I turned it on again. The battery is supposed to hold its charge for 15 to 20 days before needing recharging but I refuse to leave my ST off long enough to test that. If I go on a long vacation, (and don't bring my ST), I'll find out if it lasts that long. I have had the ST off for twelve to fifteen hours and the time/date was still correct.

Also included with the TIME-SAVER is a program to display the time and date when you turn the computer on. This program, "Timesave", should reside in an "AUTO" folder on each disk you boot from. When the system is turned on or reset, the control panel resets the displayed time/date to the one originally set in ROM. "Timesave" resets it to the current time/date. If "Timesave" isn't run, the time/date displayed in the control panel will be incorrect but the correct time/date is still stored in the TIME-SAVER so the next time "Timesave" is run,

the panel will be updated correctly. There are later versions of the control panel that don't update the time/date when the system is reset, so, if you use one of these, it isn't necessary to run "Timesave" for the time/date to be displayed correctly.

There is another benefit to TIME-SAVER. If you use the @TODAY function in VIP, the control panel must be loaded as a desk accessory and takes up memory which should be available to your spreadsheet. Instead, delete the control panel accessory from your disk and run or auto-run "TIMESAVE" before loading VIP. I did this and had the correct date plus an additional 17616 bytes available. Wow! A free 17K upgrade included with every TIME-SAVER!

The TIME-SAVER is a fine product and I think it's more than worth the \$49.95 retail price. MIND MINE also markets their 1 MEG 520ST upgrade, (reviewed in April 1986 ACE), MT-Forth-83, talking personalized puppets (I've got to see these), educational VHS tapes and many other products. They seem always willing and eager to answer questions and to help in case of any problems. It's good to know that they support the products they sell.

— Steve Golden

TYPESETTER ST and RUBBERSTAMP ST

XLENT SOFTWARE has released ST versions of these programs to meet the graphic and text needs of various programmers on the 16 Bit machines. Both of these programs have a similar layout and contain some very powerful features as well as a wealth of available text options.

RUBBERSTAMP allows the user to shrink, expand, rotate, and perform a number of options hitherto unavailable. To do this, however, takes many hours of practice and a lot of trial-and-error sessions.

Because of this, I find the programs to be slow, cumbersome and not very predictable for my uses. For example, shrinking a screen of text down usually means it becomes no longer legible. Likewise, my attempts to create icons were not very successful.

I feel these programs are not very user-friendly. Even after about 8 hours of experimentation, I find myself chained hopelessly to the manuals. Most all action occurs through manipulation of the function keys. While this seems straight-forward enough, in practical applications the "keys" do different things in different modes. I am constantly destroying my creations thru the confusing multi-purpose function keys.

Perhaps through constant use, one could really take advantage of these programs and use the power they are probably capable of demonstrating. I find I prefer to use less powerful but more user-friendly approaches to graphic/text manipulation.

In closing, it may be important to note these applications are merely ported over from 8-bit programs and do not attempt to interface with the Gem desktop at all.

— Graham Smith

vp ramblings

We are now into the month of July and we are into the last half of the year. Not many new things have come along for the ST's but the 8-bit stuff has really been few and far between. I keep hearing about new things that will be along very shortly, but they fail to materialize or they just aren't up to what I thought they would be. Being the eternal optimist I keep hoping something will turn up. I hope this doesn't mean the 8-bit is dead, but even if it does there is still enough stuff out there to keep one going for quite sometime. If the 8-bit dies it doesn't mean computing is dead it only means we will go on with the newer machines, also I'm sure someone will come up with an 8-bit emulator for the ST. I think whoever does will make a lot of money with it.

We are going to bulk mail this month and we hope you will send in the coupon found elsewhere in the newsletter to let us know when you received the newsletter so we can gauge what time of the month we need to send out the newsletter so everyone can receive it for the month it is written. As you know bulk mail is one of the last to be delivered and we do want you to get it the same month we send it. Thank you for your help with this.

The newsletter should get bigger in size and later on in the number of pages, so we can give you more features, programs, information that everyone will get something more from it.

— Larry Gold

PICKS OF THE YEAR

While there is some rumbling about the inevitable extinction of the 800/130XE series of computers and an accompanying drop in software for these computers, there is also evidence the quality of software is at a high-water mark.

My picks in the gaming world of newer introduction, are ALTERNATE REALITY Datasoft and Cygnus' introduction of STARFLEET I. Both of these games are strategy games with great depth of content. They are the type of games which sell computers, but neither is easily mastered. With patience, however, the player gratification is high and there is considerable excitement generated by multi-action gameplay.

ALTERNATE REALITY is a role-playing graphic/text adventure game which is essentially a dungeons and dragons discovery game. It holds your interest well as you attempt to build a character thru seemingly (at times) impossible circumstances. The game is not easy to get started in but once I got going it was very addictive and contained a wealth of unpredictable climaxes. Apparently DATASOFT is already working on sequel modules and plans are underway for an ST version available around Xmasetime. I don't like adventure games but this one is a classic in playability and interest. Definitely 10 stars!

STARFLEET I is a strategy game of outer space liberation. What sets it above other space games is the play-quality, superb documentation, and all the little touches which stand out "above the crowd". There are over 100 pages of documentation necessary before all the details come together, but the effort is well worthwhile. I found myself totally absorbed. What a great Sunday afternoon escape! Hopefully there will also be an ST version available soon as the game is a natural for "the mouse".

— GrahamSmith

STuff

ST World and **ST Retail News** are a couple of new ST magazines which are offered free to "qualified" subscribers. Write them for a subscription blank to see if you're qualified: CERC, Inc., P.O. Box 2310, Roseburg, OR 97470.

The Antic SIG on Compuserve has a good report on the June CES. The TT 32/32 uses the Motorola 68020 chip. The 68020 is in the same "family" as the 68000, making all 520ST and 1040ST software "downwardly-compatible" with the new machine. Atari is toying with two possible configurations: either an open architecture machine with slots, or using the ST as front end to the 32-bit as number cruncher.

THE IBM ST: According to Atari's Sig Hartmann, the ST IBM PC emulator will "hopefully" be ready before the end of the year.

THE APPLE ST: Atari is also considering the possibility of developing their own Apple II emulator. This move will allow schools with ST computers to tap the vast Apple II education software library. Hartmann is negotiating the legal complications with Apple Computer Corp.

BLITTER CHIP: Questions on the rumored "blitter chip", said to improve the ST's graphic capabilities, prompted Hartmann to announce, "We will definitely have a one-million pixel machine early next year." It is not known if he was referring to an add-on for the existing machines, or the resolution of the new 32-bit workstation.

Hartmann also vaguely mentioned a project to use the ST as the "brain" for a low-cost laser printer.

80-COLUMN CARD ADDENDUM: For the technically minded who are wondering how to program the Atari 80-column card, Jose Valdez of Atari tells us the adapter takes E: device calls and can also receive P: device calls. Just send certain codes to the device and you're on your way. The card will be completely "transparent" to any software using the E: device — such as BASIC cartridges. Programs addressing the screen directly will run into some unusual problems and have to be reprogrammed to work with the new card.

A-MAZE-ING! Xanth, creator of the 8-bit and ST Boink! and Fuji Boink! demos is developing a 3-D maze game with smooth-scrolling mazes. The trick here is that Xanth plans to make this a multi-machine game where each player can track down another with the maze. That's all we have from Antic.

A lot of the rumors this month have to do with emulators. New products for the ST are appearing in a virtual avalanche, and yet emulators to run other software is HOT news. The Mac cartridge appears to be on the near horizon. Latest rumor has it the Smalls have successfully negotiated a deal with Apple. The writing was on the wall. The cartridge was apparently going to appear — deal or no deal. If no deal, the cartridge would be bare of the Mac ROMs (with

instructions on how to go to your local Apple dealer and buy the ROMs). \$99 for the bare cartridge, \$299 for the complete cartridge. How about a 512k Mac and an Atari ST, both for less than \$1000 total!

ST Library

QuickView Systems has sent us a demonstration version of their popular ZOOMRACKS program. It is apparently fully implemented, except you can only manipulate up to 20 records in any file. This is a good way for you to see if the program will suit you. Our Utility disk has the following additions: CRASH.PRG reports the number of bombs (or mushrooms) on the last crash, and describes the error. It includes values for registers and stacks; SHOWCODE.TOS reports the internal code of any keypress on the ST keyboard; FMT10MNO.TOS, FORMAT10.TOS and .DOC will format disks with 10 sectors per track (adding 40k single, or 80k double-sided storage); MKCRLF.TOS forces a carriage return and line feed for the end of line character; SECED.PRG & .DOC no frills sector editor; STSQU.TTP & .DOC, STUSQ.TTP are squeeze and unsqueeze utilities. The RAMDISK folder now contains: FRAM.S 68000 assembler source for a 720k ramdisk; RAMDISK.PRG & .DOC by John Harris is a variable ramdisk can be configured to autoload and size in increments of 64k. Survives warmboot; RAMDSK1M.PRG for 1-meg STs, survives warmboot.

Our Desktop Accessory disk has the following additions: INTRAM.ACC & .RSC survives warmboot, size is set from desktop and can be RESIZED without continually eating memory as you allocate and deallocate ramdisks; (In the Pascal/Modula-2 folder) XBIOSCALL.TXT and .SRC — Modula-2 xbios call routines; SCRSAV.ACC color monitors go black after a few minutes; mono screens flip each color every 8 seconds. Any keystroke disables it; SETPALET.TOS sets screen colors from data in the DESKTOP.INF file, saving you the memory cost of the Control Panel. When you get it the way you want it, place it into an AUTO folder before a COMMAND.PRG shell; QDATE.PRG used in an AUTO folder. It remembers the last date the system was booted. It's used to reset the date and shows the day of the week, too; MEDRES.PRG in an AUTO folder will default color boot-ups to medium resolution instead of low.

Our C-Disk has the following adds: LIST.TTP & .DOC adds line numbers to the beginning of each line of a C program, can also output a line numbered program to disk; MIKERAM.C source code for graphic-based ramdisk desk accessory for you to modify to your heart's content.

Our Word Processing disk has FILTER.TOS which strips all non-USASCII characters from any text file.

We have a new disk we call LIBRARY which contains disk indexing, directory search, and directory listing and printing utilities.

JULY INSOFT

The July, 1986 issue of InSoft (1834 Beacon St., #1, Brookline, MA 02146) magazine contains a good selection of news, executable programs and document files. This issue contains a new version (0.6) of Neo-Chrome which implements some features in all but one of those empty boxes we've been wondering about. One feature is an "animation" feature. Also included are some good C programming utilities.

InSoft also sends the following items to those who join their ST Network: a product line catalog; lists of hardware and software specials every 2 months; membership number to process your orders.



ST BENCH

The following benchmarks were done on Megamax, Lattice, and DRI on the Atari ST. Because the results were so strikingly different, I also included results for DRIE which is DRI with my own runtime written entirely in assembly language.

All benchmarks were run on a 520 expanded to 1 meg. I used MDISK from Michtron for the ramdisk, because it is the fastest I've tested. I used the command shell from DRI, and created a batch file for each compiler to compile and link in a single line command.

SIEVE is the standard prime number test. FIB tests function recursion. INTMATH tests addition, subtraction, division, and multiplication. POINTER tests pointers. STRTEST tests the functions strcpy(), strcat(), strlen(), and strcmp().

SIEVE (word)	DRI	DRIE	MEGAMAX	LATTICE
comp/link	49.3	19.5	12.4	20
size	10,325	2,786	6,813	22,776
run time	4.5	4.5	4.2	5.2
SIEVE (long)	DRI	DRIE	MEGAMAX	LATTICE
comp/link	49.3	19.5	12.4	20
size	10,331	2,792	6,823	22,790
run time	5.3	5.3	5.2	5.4
SIEVE (register)	DRI	DRIE	MEGAMAX	LATTICE
comp/link	49.3	19.5	12.4	20
size	10,269	2,730	6,773	22,766
run time	2.7	2.7	2.6	4.2
FIB	DRI	DRIE	MEGAMAX	LATTICE
comp/link	47	18.5	12.1	17.9
size	10,273	2,734	6,797	14,558
run time	16.7	16.7	19	19.8
INTMATH (word)	DRI	DRIE	MEGAMAX	LATTICE
comp/link	52.8	23.5	12.7	55.7
size	10,687	3,148	7,199	14,689
run time	3.9	3.9	4.0	2.8
INTMATH (long)	DRI	DRIE	MEGAMAX	LATTICE
comp/link	53.9	23.5	12.7	25.3
size	10,959	3,260	7,433	14,764
run time	11.3	9.7	14.3	19.6
INTMATH (register)	DRI	DRIE	MEGAMAX	LATTICE
comp/link	52.8	23.5	12.7	56
size	10,406	2,867	6,923	14,667
run time	2.7	2.7	2.5	2.5
POINTER	DRI	DRIE	MEGAMAX	LATTICE
comp/link	46.6	16.8	12.1	17.1
size	10,139	2,600	6,665	14,402
run time	13.6	13.6	18.9	14.9
STRTEST	DRI	DRIE	MEGAMAX	LATTICE
comp/link	44	18	10	15.5
size	8,132	2,815	3,718	11,131
run time	16.6	12.2	15.6	45.6

CONCLUSIONS:

Lattice produces by far the largest code. It generally compiles faster than DRI, but slower than Megamax. The code it generates is usually slower than DRI, but faster than Megamax. Its worst showing was in compiling the INTMATH program, though it ran it the fastest. Otherwise, Lattice is easy to use, and provides excellent error messages and warnings. However, its code size is quite excessive. It would be a terrible compiler to use to write accessories. It does use buffered I/O, so is an excellent compiler for file handling if you don't mind the program size.

Megamax is by far the easiest compiler to use, and generates the smallest code. With the exception of the Sieve, the runtimes are usually slower than either DRI or Lattice. It provides you with the ability to run it from a shell, or from their own gemtop environment. Both work extremely well. If you're not planning on writing commercial software, this is definitely the best compiler to get.

DRI consistently generates the fastest code. The comparison with DRIE shows its main weakness - the poorly written runtime library which bogs down the linker and produces inefficient code. DRI is the only compiler to produce assembly language output which can then be optimized by hand. Furthermore, DRI can produce a program with symbols, and you can trace through a program using their symbolic debugger - assuming you know assembly fairly well. DRI is well suited for commercial software development with advanced programmers who will be doing a lot of optimization, and writing many if not all of their own runtime routines. If you're not a professional programmer, this would not be a good compiler to use.

Note: Atari is releasing an update to DRI known as the Alcyon compiler. This compiler is slightly slower, generates slightly larger code, and produces equivalent run times. The compiler and assembler files are 64K larger.

— Reviewer: Ralph Walden

Atari Computer Enthusiasts

The ACE Newsletter is the first Atari user group newsletter to be published anywhere in the world. We began publishing in April, 1981. A.C.E. is an independent, non-profit and tax-exempt computer club and user group with no connection to Atari Corp. We are a group interested in educating members in the use of Atari computers and publishing the latest News, Reviews and Rumors.

All our articles, reviews and programs come from you, our members.

Our membership is world wide. Membership fees include the A.C.E. newsletter (10 issues per year) and high access in the A.C.E. Bulletin Board System. Dues are \$14 a year for U.S., and \$24 a year overseas airmail.

Subscription Dept.: 3662 Vine Maple Dr., Eugene, OR 97405
President — Dick Barkley, 2907 Wingate, Eugene, OR 97405
Vice President — Larry Gold, 1927 McLean Blvd., Eugene, OR 97405
8-Bit Librarian — Chuck & Jody Ross, 2222 Ironwood, Eugene, OR 97401
ST Librarian — Jim Bumpas, 4405 Dillard Road, Eugene, OR 97405
Editors — Mike Dunn, 3662 Vine Maple Dr., Eugene, OR 97405
Jim Bumbas, 4405 Dillard Road, Eugene, OR 97405
Larry Gold, 1927 McLean Blvd., Eugene, OR 97405
E.R.A.C.E. (Education SIG Editor) — Nora Young, 105 Hansen Lane Eugene, OR 97404 / (503) 688-1458

Send 50¢ stamps or coin (\$1 overseas) to the Librarian for updated ACE Library lists.

A.C.E. Bulletin Board

(503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a 130 XE upgraded to 320k by Charlie Andrews, 2 double-density disk drives, and a 10-meg hard disk from Supra Corp, an Epson MX80 printer, a 2400 US Robotics Courier modem, running the PD/M Bulletin Board software.

Best of ACE books

Volume 1 contains bound issues of the ACE Newsletter from the first issue, Oct. 1981 to June, 1982.
Volume 2 contains July 1982 through June 1983.
Only \$12 each (\$2 extra for airmail). Available only from George Suetsugu, 45-602 Apupau St., Kanoehie, HI 96744.

SortFinder Index

A composite index of Atari related articles in four popular computer periodicals, including the ACE Newsletter. Volume 1 covers April, 1981 through June, 1983. Volume 2 covers July, 1983 through December, 1985. Only \$6 per printed copy or \$11 per disk copy for ACE members. Available from: Jim Carr, 2660 S.W. DeArmond, Corvallis, OR 97333.

30 East 13th Avenue **Eugene, Oregon 97401**
Phone 503/683-2657

PUBLICATION DESIGN, EDITING, & PRODUCTION

Editing & Design Services

ATARI OWNERS: If you have a modern, text editor, and communications program to send ASCII files, you should consider the improved readability and cost savings provided by TYPSETTING your program documentation, manuscript, newsletter, or other lengthy text instead of just reproducing it from line printer or daisy-wheel output. Computer typesetting by telephone offers you high quality, space-saving copy that creates the professional image you want! Hundreds of type styles to choose from with 8 styles and 12 sizes "on line." And it's easy to encode your copy with the few typesetting commands you need.

COMPLETE CONFIDENTIALITY GUARANTEED

— Bonded for your protection —



BULK MAIL
U.S. POSTAGE
PAID
PERMIT NO. 472
EUGENE, OREGON